

FIGURE 1
(Prior Art)

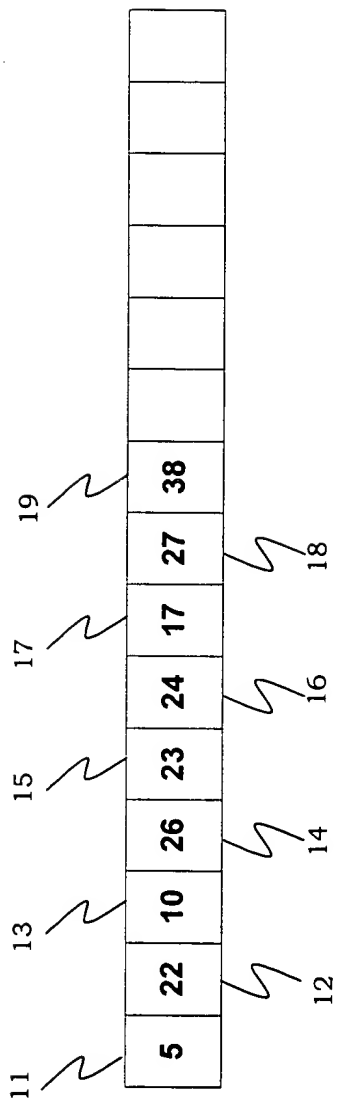


FIGURE 2
(Prior Art)

A) The value in the root node is removed, leaving a "hole".

C) The new value for root node is compared with the values in the root's two children nodes. The value of these three with the highest priority is placed in the root node, and if this means that one of the children's values must be moved up, then this is done and the child is then considered a root of a subtree so that the process can be repeated down the heap until the bottom of the heap is reached or until no more swaps are necessary.

B) The value in the bottom-most, right-most position of the heap is selected and is moved to the root node. (I.e., the last value in the heap is moved to the top of the heap).

FIGURE 3
(Prior Art)

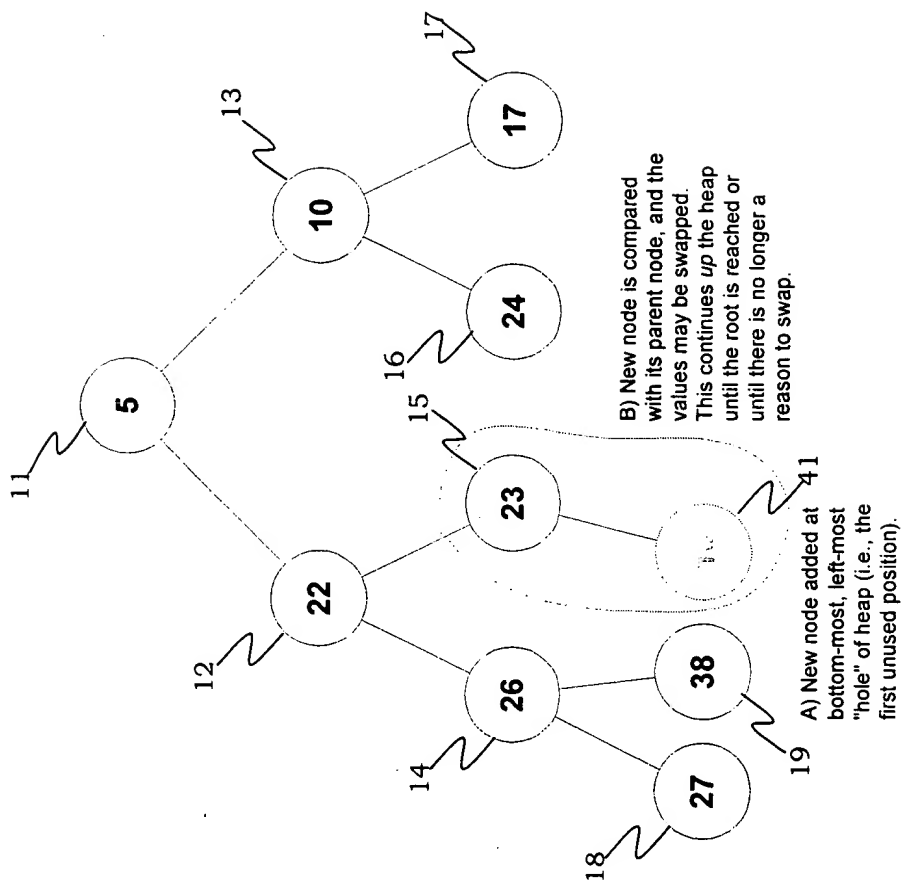


FIGURE 4
(Prior Art)

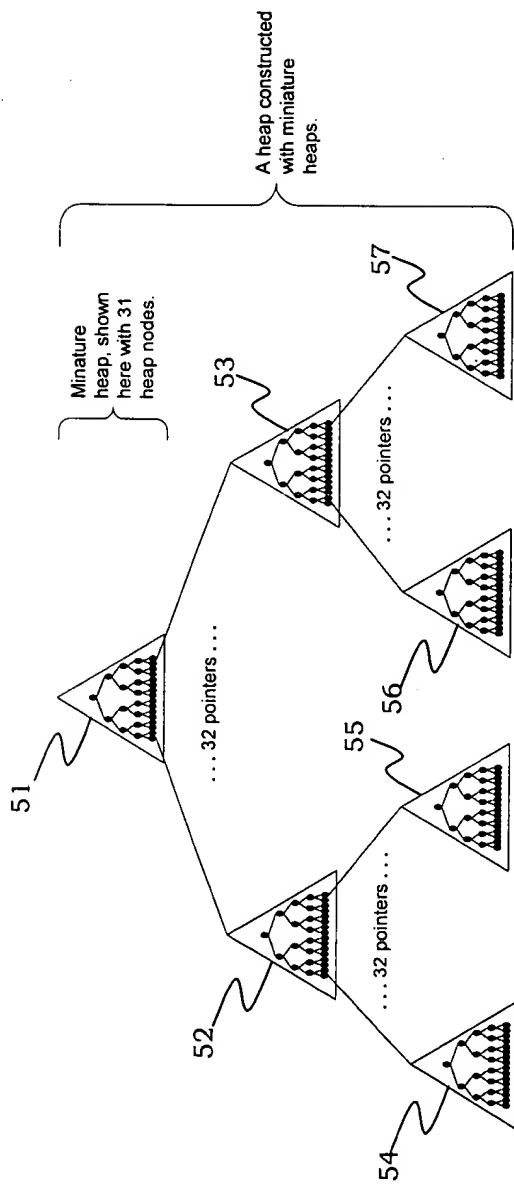


FIGURE 5

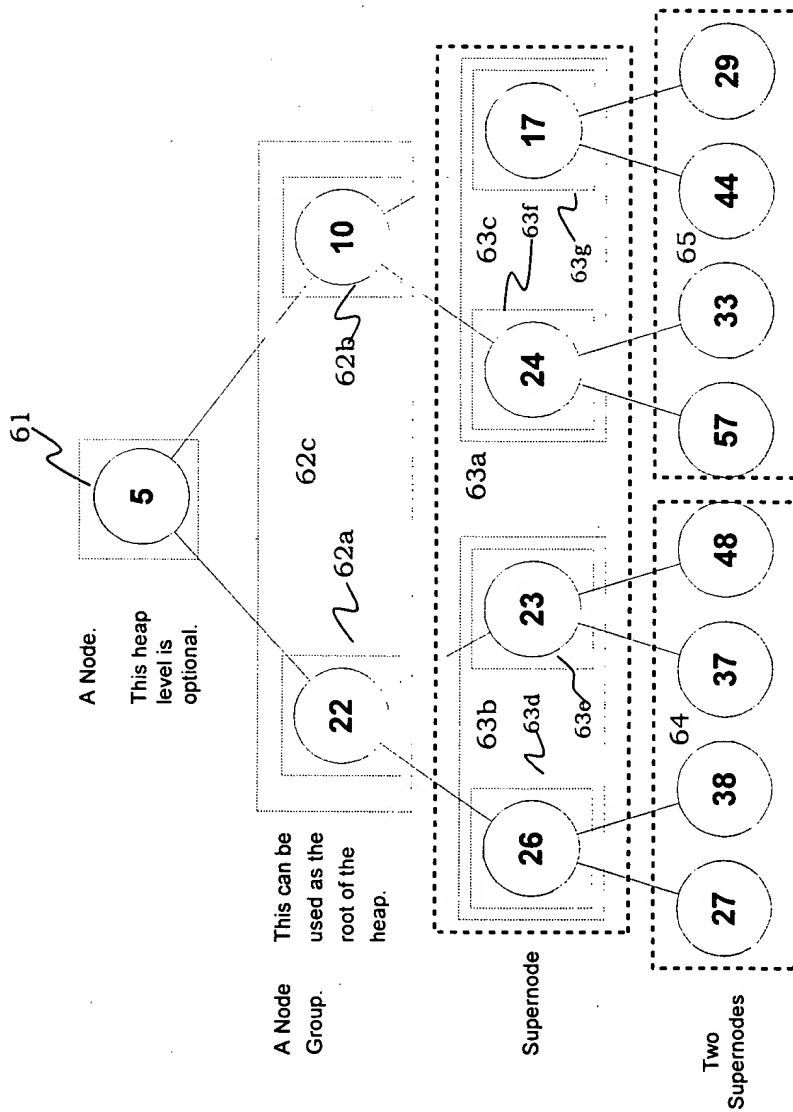
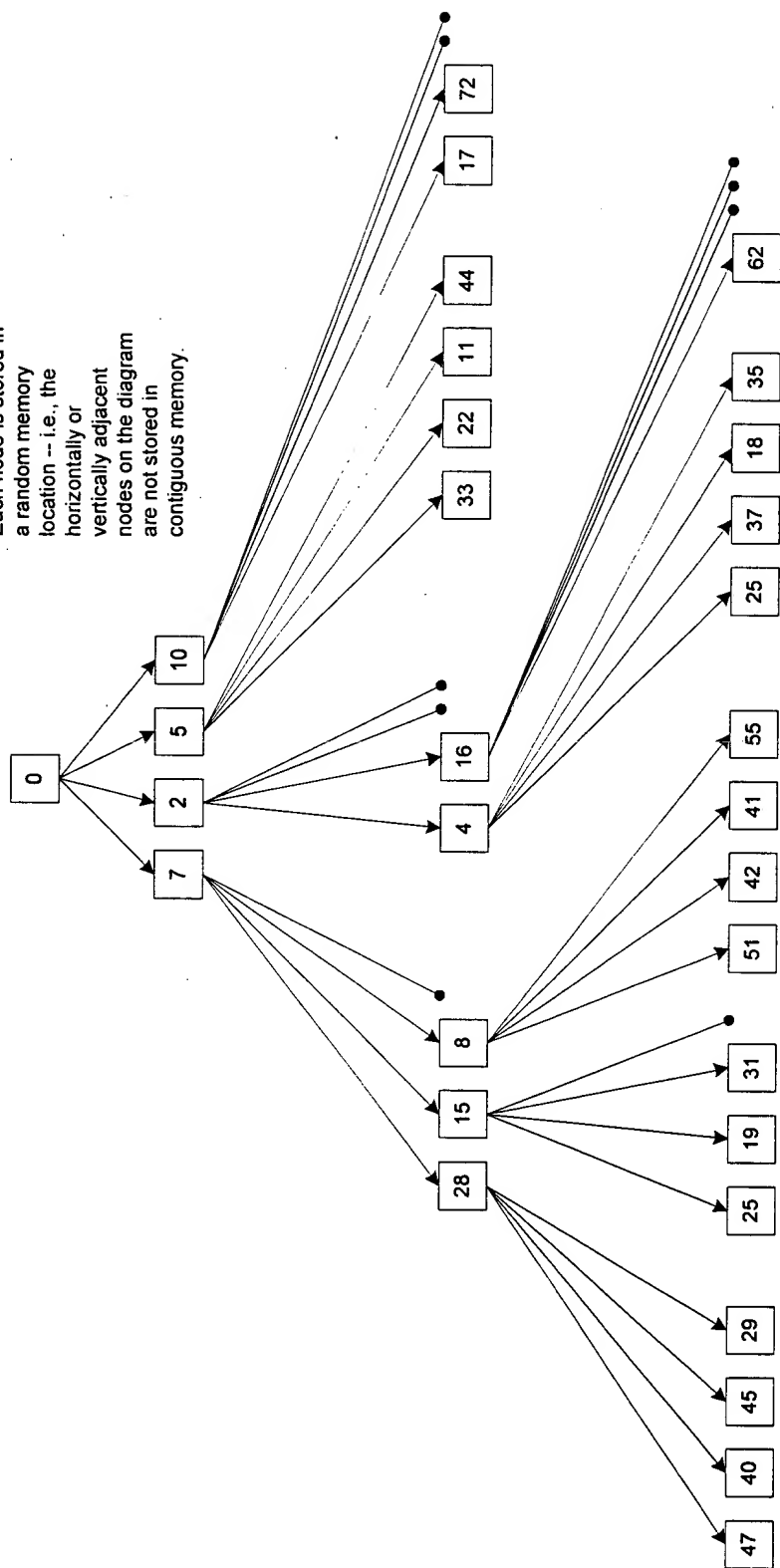


FIGURE 6

Each node is stored in a random memory location -- i.e., the horizontally or vertically adjacent nodes on the diagram are not stored in contiguous memory.



70

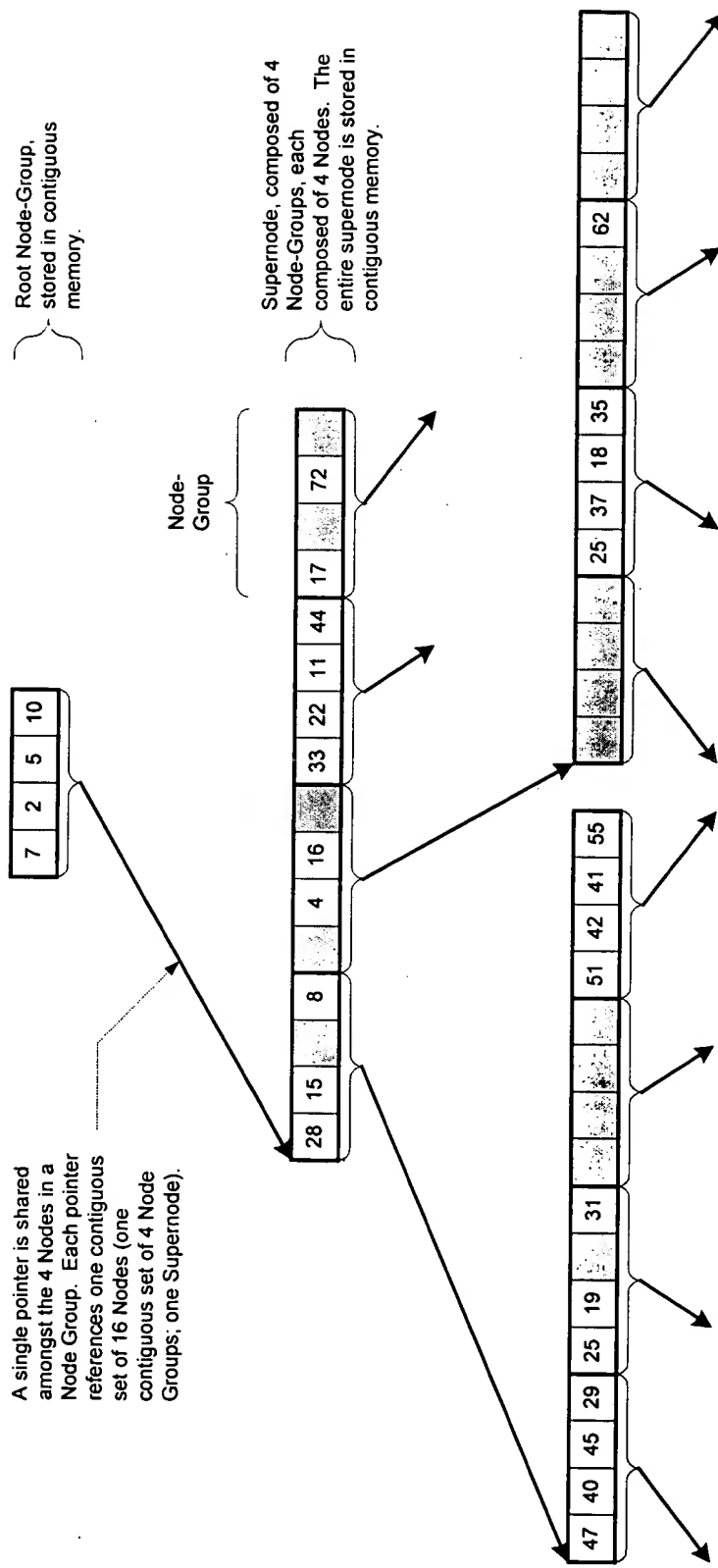


FIGURE 8

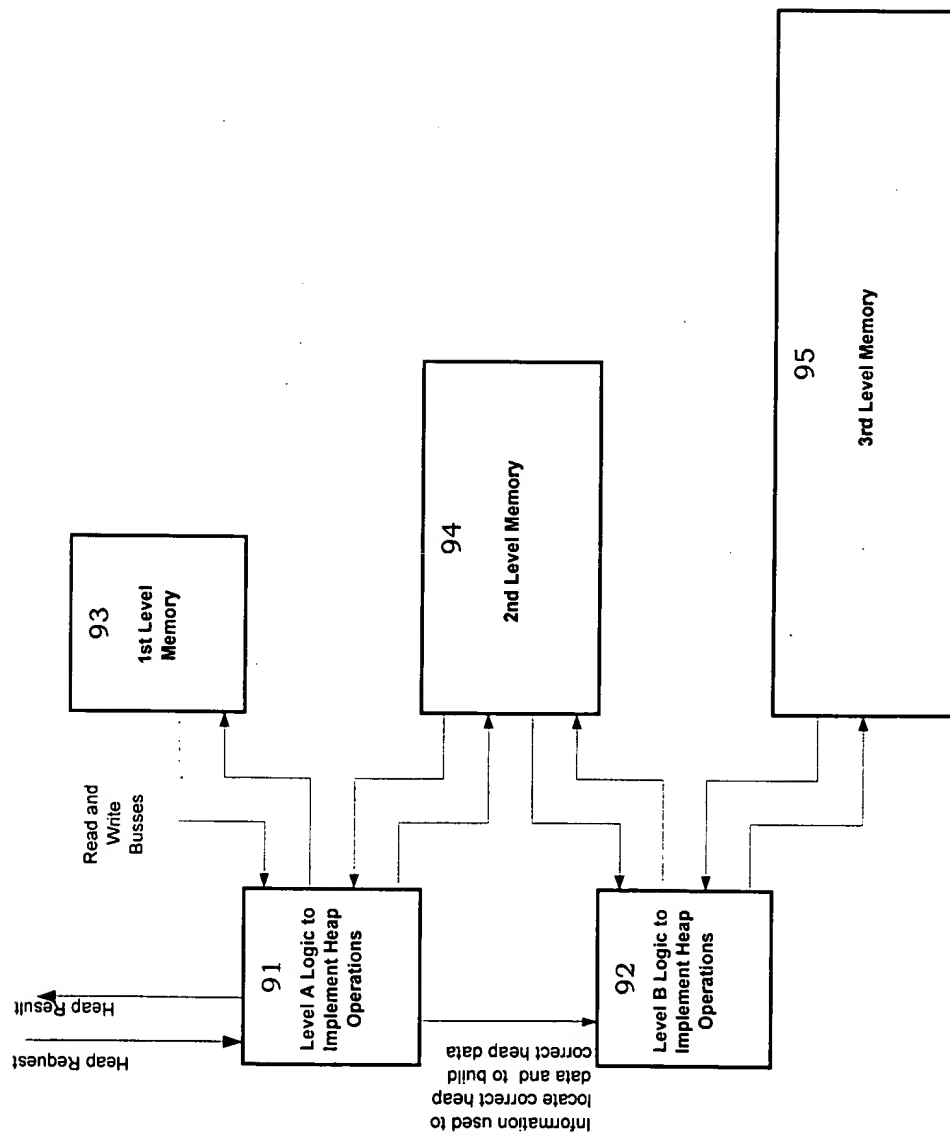


FIGURE 9

	time ----->																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Read Level 1 RAM	A						B											
Write Level 1 RAM					A							B						
Level A Comparisons				A	A					B	B							
Read Level 2 RAM			A						B									
Write Level 2 RAM							A							B				
Level B Comparisons					A	A						B	B					
Read Level 3 RAM					A						B							
Write Level 3 RAM										A							B	
Level C Comparisons							A	A						B	B			
Read Level 4 RAM							A						B					
Write Level 4 RAM										A						B		

FIGURE 10

2025 RELEASE UNDER E.O. 14176

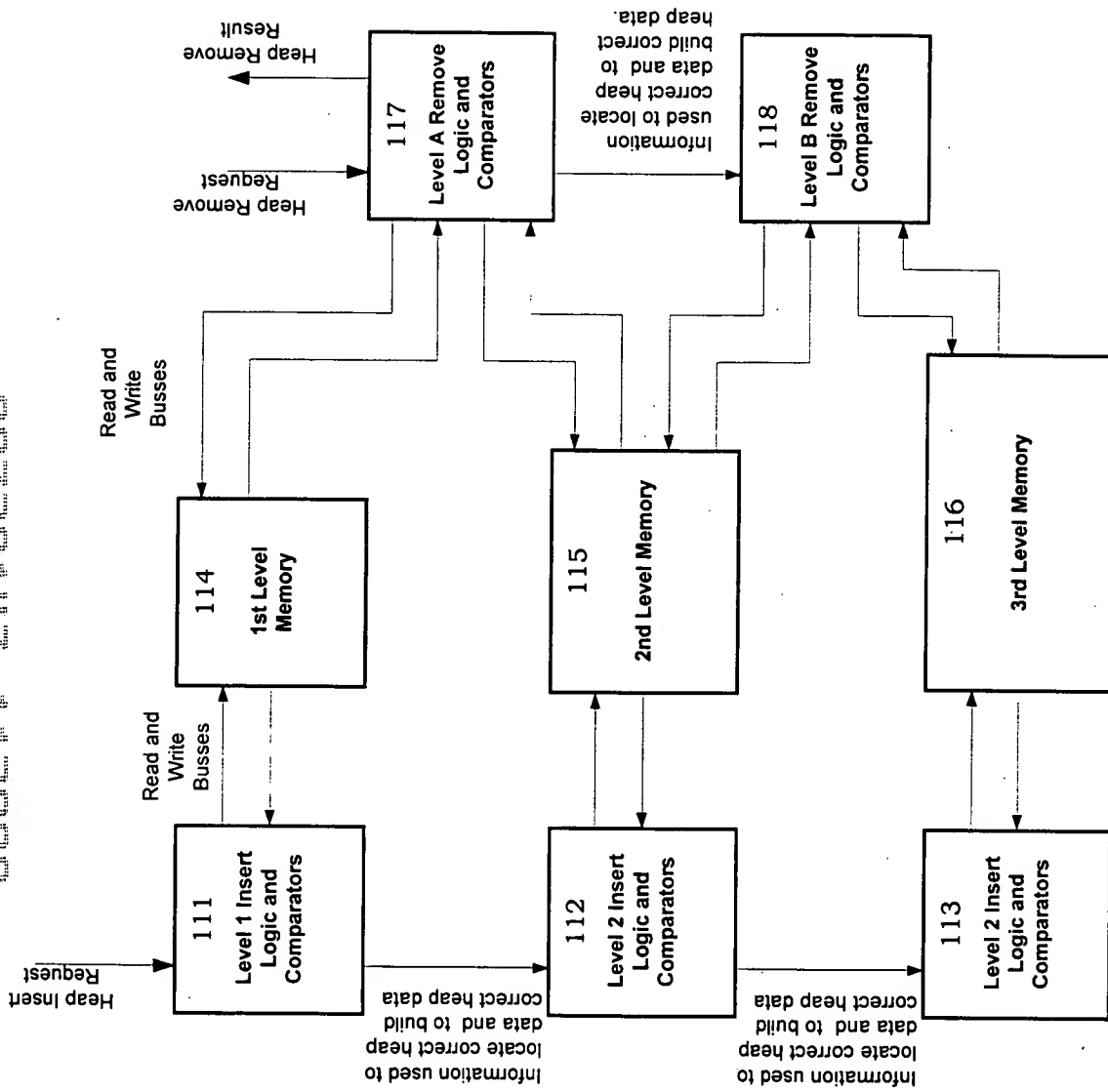


FIGURE 11

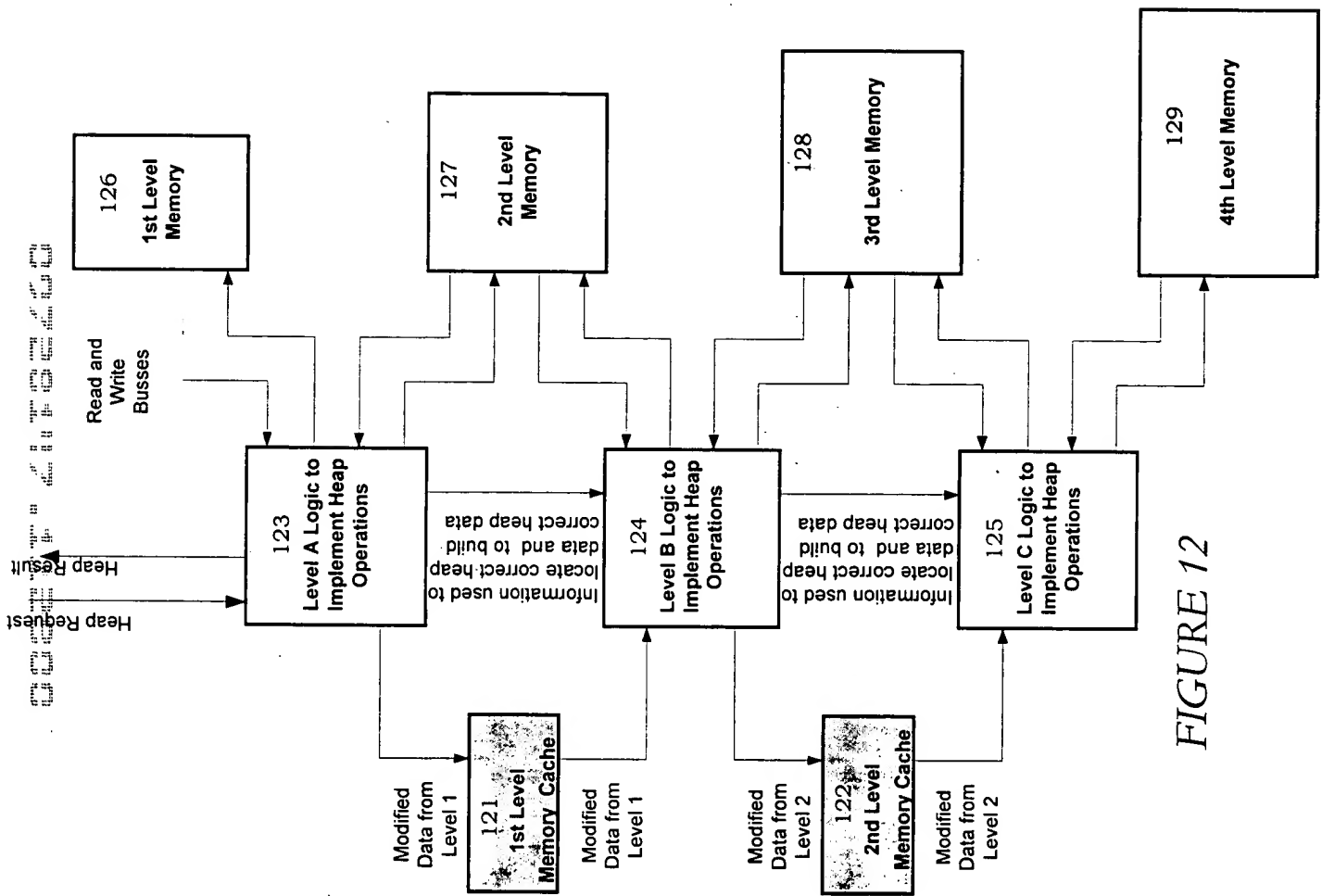


FIGURE 12

I. The 2nd request (B) starts to read from level 1 memory. The 1st request (A) starts to modify the data that it read from level one. Thus, if request B reads from the same location as request A, request B will get the old (stale), unmodified data and produce the wrong result.

II. Once request A has finished modifying the data that it read, it can write it back. However, it must also cache the data so that when B starts to modify the data that B read, B can discard the data it read and use the current information in the cache instead.

III. Request B checks the cache before it modifies the data that it read. If the cache indicates that B is operating on the same memory location in level 1 that request A just operated on, B uses the contents of the cache.

	time →									
Read Level 1 RAM	A									
Write Level 1 RAM										
Level A Comparisons										
Read Level 2 RAM										
Write Level 2 RAM										
Level B Comparisons										
Read Level 3 RAM										
Write Level 3 RAM										
Level C Comparisons										
Read Level 4 RAM										
Write Level 4 RAM										

FIGURE 13